Regenerative Braking in Electric Bike Conversion Kit

By

Chloe Armstrong

Jace Haas

Lucas Pruett

Final Report for ECE 445, Senior Design, Spring 2023
TA: Matthew Qi

03 May 2023
Project No. 38

Abstract

The regenerative braking e-bike kit shows that a simple, cost-effective e-bike kit can be designed with the capability to recover some of the kinetic energy of the bike when braking. This kit enables a regular bicycle to be moved forward with a brushed DC motor, and it can apply stopping power with regenerative braking. This kit failed to properly integrate forward motion and regenerative braking due to design issues that could have been resolved given ample time.

Contents

1. Introduction

The popularity of electric bikes has risen over the last decade. These types of bicycles provide a greener alternative to cars, as well as a faster method than traditional bikes to traverse urban areas. However, electric bike's limited range may discourage some from taking the leap. An average electric bike has a range of around 20 to 40 miles on a single charge [1]. This range may not be worth the upwards of $1000 price tag for an entry level product [2].

Electric bikes also suffer from the same brake wear as a traditional bike. It has been estimated that traditional cyclists replace brake pads every 500-1000 miles [3]. Not only can brake maintenance be seen as an undesirable chore, but from an engineering perspective, it also wastes perfectly good kinetic energy.

Our solution to both problems is an electric bicycle kit that utilizes regenerative braking. This solution returns otherwise wasted kinetic energy back to the battery. Since the battery is recharged during this process, the range of the electric bike is extended. Regeneration also does not require the use of manual brakes. Since these are used less, the need to frequency brake maintenance is reduced.

The project is divided into three main blocks: power, motor, and control. Figure 1.1 shows an overview of how these blocks interact, as well as the primary components within each block.

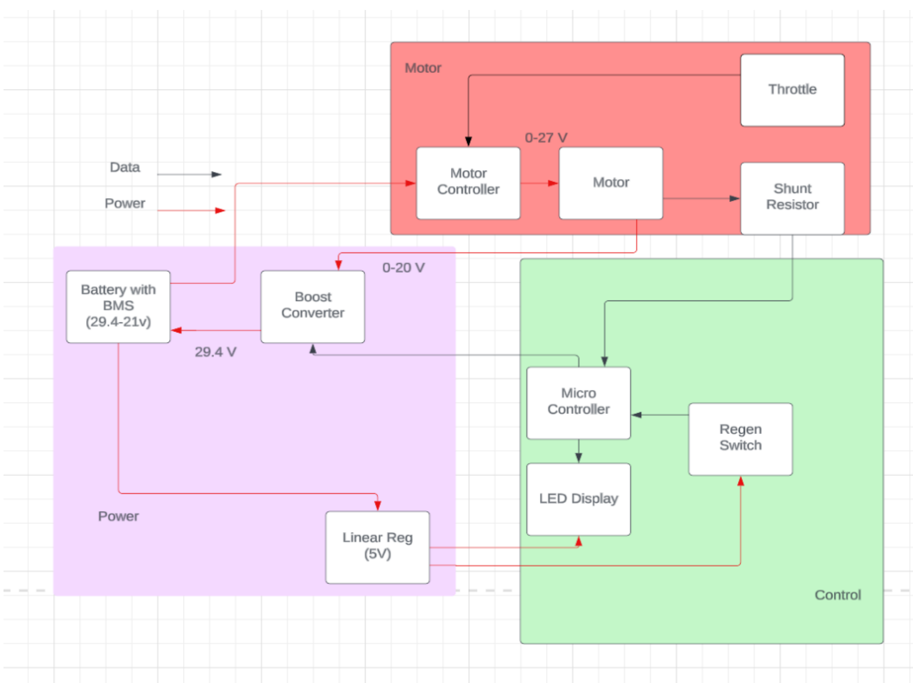


Figure 1.1 Block Diagram

The power block includes the 24 V lithium-ion battery needed for bike power. It also contains the boost converter necessary for regenerative braking, which will be discussed in more detail in Chapter 2. The power block supplies power to all bike components and receives the energy regenerated during regenerative braking.

The motor block includes the motor, motor controller, twist throttle, and shunt resistor. It is primarily responsible for controlling acceleration and deceleration while the bike is being driven forward by a throttle system controlled by the user. This will be referred to as “forward mode” going forward. The shunt resistor is used to monitor current produced by the motor when the user has placed the regenerative braking switch in the on position. In what will be referred to as “regenerative mode”, the throttle is disabled, and the bike is using recovered kinetic energy to charge the battery.

The control block is responsible for all control signals sent to other blocks. These signals will be discussed in more detail in Chapter 2. The bike mode, which is determined by the regeneration switch, determines the value of these control signals. The control block also processes the voltage data seen at the shunt resistor and displays this processed data on an LCD screen that can be viewed by the user.

There are a few higher-level performance requirements specified for this design. More detailed requirements and verifications will be discussed in Chapter 3. The primary requirement is that regenerative braking must be able to recharge the battery to some extent. Pressing a switch on the bike must initiate regenerative braking in the bike. The stopping power of this braking must be able to stop the bike within a reasonable distance. For example, if the bike was traveling at 15 miles per hour, it must come to a stop within 150 yards.

In the forward mode, the motor must be able to accelerate the bike to 10 miles per hour or above. There must also be a way to disconnect the battery from every other component on the bike in case of an emergency.

Though the blocks remained the same in name throughout the project, the components they contained and the functions they performed were modified throughout the course of this project. These changes were primarily aimed at increasing the simplicity of the design. Figure 1.2 shows the original block diagram presented in our initial Design Document.

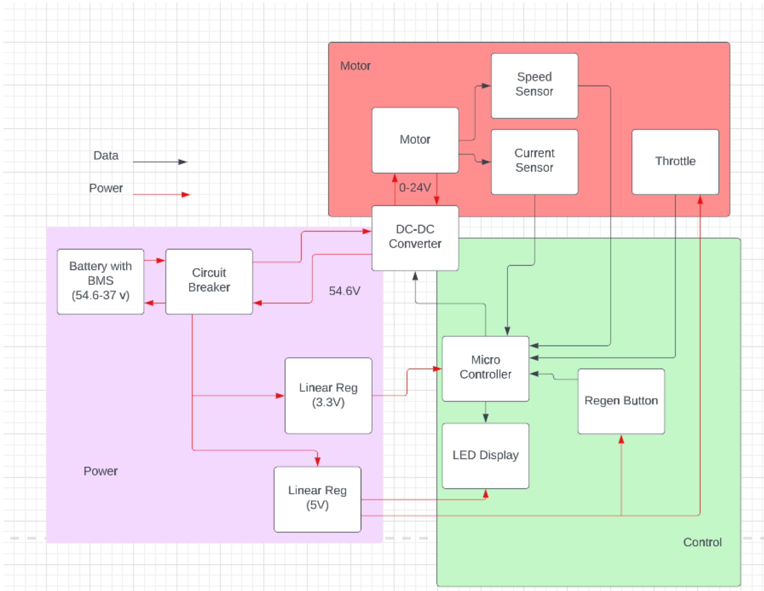


Figure 1.2 Initial Block Diagram

Much of the process behind the changes shown in Figure 1.2 will be discussed in Chapter 2. The most impactful change is the decision to implement a motor controller and boost converter as opposed to one bidirectional DC-DC converter. The battery voltage was also changed from 48 V to 24 V to increase the switching converter efficiency. The speed sensor, 3.3 V linear regulator, and the circuit breaker were eliminated from the final design, as these were deemed unnecessary for implementation.

2 Design

2.1 Design Procedure

2.1.1 Power Subsystem

Our part in the design of the power subsystem was focused on the interaction between the lithium-ion battery (LiPo) and the motor. The idea was that in one mode the motor would consume power from the LiPo and turn it into mechanical energy. Then in another mode the motor would take kinetic energy from the momentum of the bike turning it back into electrical energy.

In the second mode the motor would behave as a mock generator. In the first mode, the power would flow from the LiPo, to the electronic speed controller (ESC) and into the motor. In the second mode, the power would flow from the motor. The motor, now acting as a generator, would distribute power to the custom boost converter, which would step up the voltage to charge the LiPo. Lacking experience with boost converters, we used the Texas Instruments WEBENCH Power Designer to create a boost with our desired functionality.

Originally, we intended to use a buck-boost converter to manage this two-way power flow. However, this increased the complexity of our design significantly, and most converters we could find weren't rated for the voltages we were using. Furthermore, modulating the output of a buck converter to control motor speed wasn't feasible, and we ultimately decided to use an ESC for driving the motor, and the boost for returning power to the motor.

Due to the two different modes of operation, a protection circuit was needed to ensure current would flow as desired. Our idea for a protection circuit consisted of two different power MOSFETs. One would disconnect the motor controller from the motor in "regenerative mode", and the other would disconnect the motor from the boost circuit in "forward mode". This design had some issues that will be discussed later, but relays could have been utilized to achieve the same effect.

2.1.2 Motor Subsystem

The motor subsystem was designed to propel the user forward at the twist of a throttle. The motor we wanted initially was a hub motor that would allow for more direct torque transfer to the rear wheel. However, due to the electrical complexities of hub motors we choose to go with a brushed DC motor. This simplified our boost conversions; however, we had to go with a motor that had a larger form factor. This meant that we could not directly mount our motor of choice to the axle. To allow for torque transfer, we chose a gravity tensioned friction wheel design.

This design would result in the motor using a friction wheel attached to the output shaft. Then, the entire motor would be allowed to freely pivot to engage the wheel of the bike with the friction wheel. The force of the engagement is based on the weight of the motor. The friction wheel design could have been improved by using a spring to make better contact between the friction wheel and the bike tire.

2.1.3 Control Subsystem

Our part in the design of the control subsystem has always focused on the microcontroller. We determined early in the project that a microcontroller would be necessary. However, the function of the microcontroller, as well as what microcontroller would be used, has changed throughout the course of this project.

The original microcontroller chosen for this project was a Texas Instruments microcontroller with part number TMS320F28035PAGQ [7]. This was chosen at a point in time where a more complicated brushless DC (BLDC) motor was being considered. The more complex buck/boost design was also still being considered at the time. At this stage in the project, we needed a microcontroller that was fit to handle the more complicated tasks that would come with these design decisions. After refining the project to reduce unnecessary complexity, this microcontroller, and the functionalities it provided were no longer needed. They would only add unnecessary difficulty to the process of programming the microcontroller. Thus, it was decided to look for a new part that better suited our needs.

After some debate, it was decided that the ATmega8 microcontroller was a good choice for this project. For one, all the team members involved had basic knowledge of and experience with Arduino IDE. This made writing the code for the microcontroller a more straightforward task. This microcontroller is also very similar to the ATmega328, which is widely used in Arduino Uno boards. This made virtual simulation of the microcontroller an option, which greatly sped up the development of the code. The ATmega8 also proved capable of the functions we were asking for. The most difficult function it was tasked with was analog-to-digital conversion, which was needed to collect data from the shunt resistor to measure the current going into the battery [12]. Finally, the ATmega8 proved to be a resilient microcontroller for this project. Throughout all our testing, and after preparing for the worst-case scenario by ordering multiple ATmega8 microcontrollers, the original ATmega8 chip we began testing with did not need to be replaced.

Though there is circuitry involved in this design block, much of the design consideration was focused on the microcontroller code in this area. An overview of the microcontroller function will be provided in this section. A more detailed look into the microcontroller and control block function will be provided in Subsection 2.2.3.

The first and primary function of the microcontroller is to interpret the position of the regeneration switch and to use that data to generate control signals. When the buck/boost design was still in consideration, we planned on using the microcontroller to generate pulse width modulated (PWM) signals for the gates in the buck/boost converter. However, with the redesign, the microcontroller only needs to supply a digital “enable” signal to our boost converter. Like many design decisions, this reduces complexity, which is a desirable trait in this case.

The microcontroller also uses the digital high or low signal from the regeneration switch to generate switching signals for our proposed protection circuit. There are two of these signals: one for forward mode and one for regenerative mode. When the mode’s corresponding signal is high, that mode is active. Since only one mode should be active at any given time, the other mode’s corresponding signal should be low. Another consideration was that there should be a delay between the switching signals in which both signals are low. This ensures protection against both modes being active at the same time. If both signals are switched at the same time, this can result in both modes potentially being active for a brief period. Since this is undesirable, protections were considered to combat this behavior.

The secondary function of the microcontroller is to measure the voltage across the shunt resistor and interpret that data. It is tasked with measuring regeneration during the regenerative mode. The LCD was not a part of our original design, but it was added to both assist with testing and to provide a measurement of regeneration to the user. The details surrounding the measurement code will be provided in Section 2.2.3. The microcontroller uses Equation 2.1 to determine current regenerated over time. Equation 2.1 reflects that current through the shunt resistor integrated over the duration of regeneration with respect to time is equal to the ampere-hours of regenerated current. In Section 2.2.3, this equation will be expanded upon.

$$Ah = \int_0^t \frac{V}{R_{Shunt}} dt \tag{2.1}$$

2.2 Design Details

2.2.1 Power Subsystem

As discussed in section 2.1.1, the power subsystem has two main components: the protection circuit and the boost converter. The boost converter was responsible for stepping up the back EMF of the motor (0-24V) and stepping it up to the battery charging voltage of 29.4V. The boost converter was designed using WEBENCH, as mentioned previously, and the final design we used is shown in Figure 2.1.

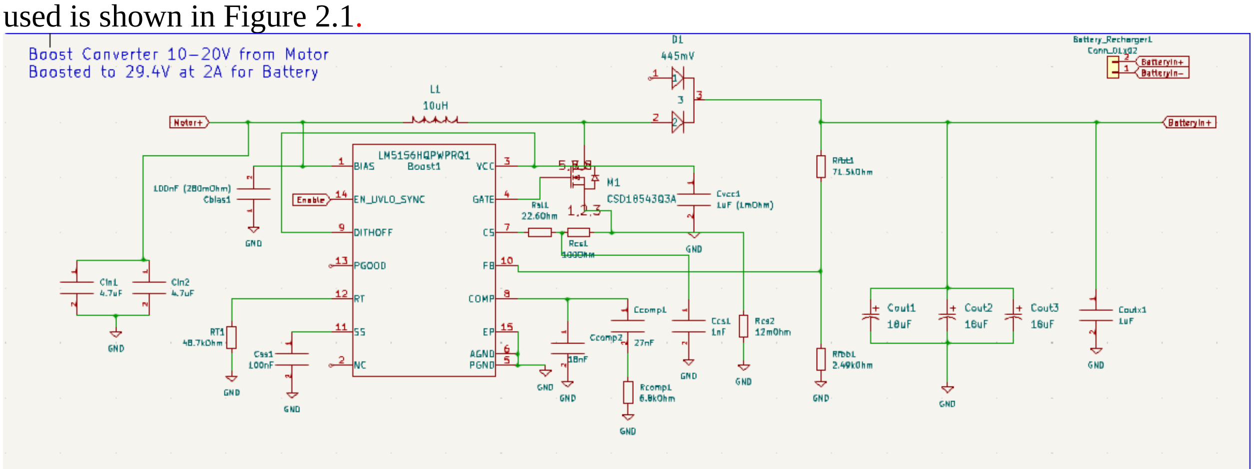


Figure 2.1 Boost Converter

The protection circuit splits the boost circuit and the ESC to ensure that only one is active at a time. This was initially designed with power MOSFETs that used control signals tied to the gates to activate and deactivate each MOSFET separately. This is illustrated in Figure 2.2. below.

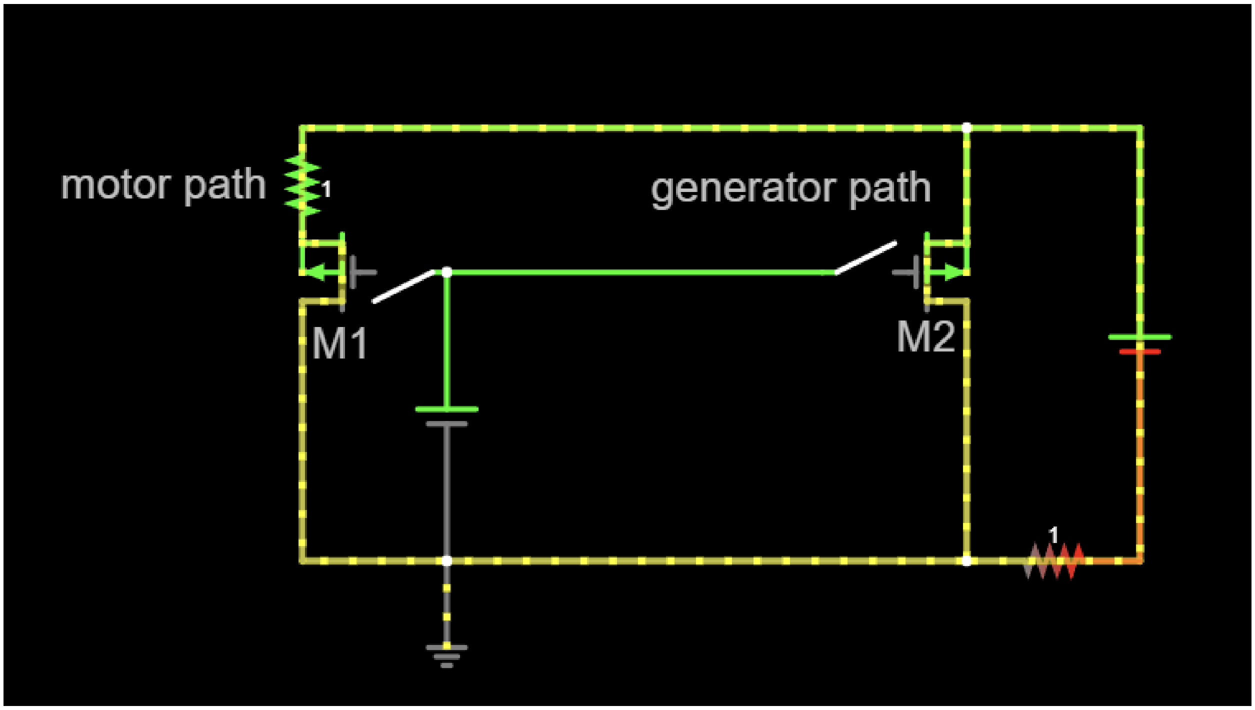


Figure 2.2 Protection Circuit

This design did not work due to the parasitic diode that bridges the source to the drain. This caused the MOSFET to leak too much current when it was meant to be closed. As a result, the motor was active regardless of the gate voltage. By the time we had noticed this issue, it was too late to redesign. To solve this issue, we should have used a relay-based design as shown in Figure 2.3.

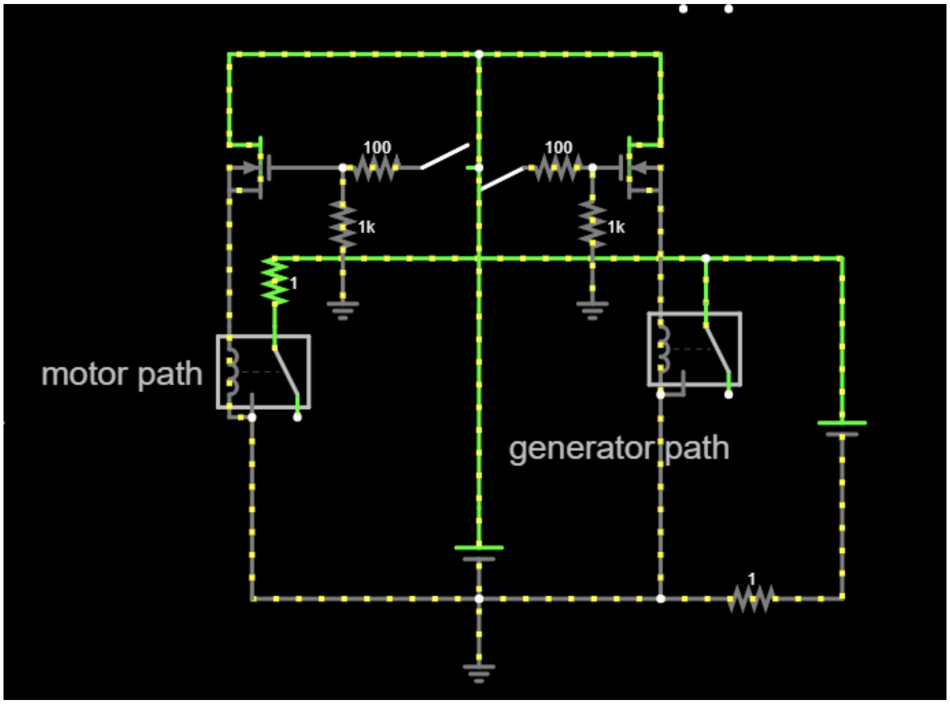


Figure 2.3 Updated Protection Design

This would physically disconnect each circuit to ensure that absolutely no current would leak when not desired. The control signals would then be attached to smaller MOSFETs that would drive the energizing the coils in the relays to open and close each relay.

2.2.2 Motor Subsystem

The motor subsystem mainly consisted of the 24V brushed DC motor, which acted as our driving force in forward mode and our power source for regenerative mode. Due to time limitations as well as lack of group member experience, we ultimately purchased a motor controller rather than designing one. This motor controller adequately drove our bike forward in response to the potentiometer throttle that came with it.

The other important part of the motor subsystem was the shunt resistor. This resistor allowed a voltage to be read by our microcontroller in order to measure the current out of the motor in regenerative mode. We used a 0.2 milliohm shunt resistor and a current sense chip with a gain of 200 in order to get a readable measurement for our microcontroller. These components allowed us to reliably measure the current from the motor without dissipating too much additional power, but we could have added controls that would prevent the current sense chip from being active in forward mode.

2.2.3 Control Subsystem

As was discussed in Section 2.1.3, the microcontroller has two primary functions. These functions, as well as the surrounding physical circuitry, will be discussed in this section. The microcontroller will primarily be discussed, but a brief description of the LCD will also be provided near the end of the section.

The input and output signals to and from the microcontroller are shown in Figure 2.4. The 5 V signals represent power to the microcontroller. The “Reg Switch” signal represents the input from the regeneration switch, which is accessible by the user. This signal can be either high or low, with a high signal representing the bike being in regenerative mode.

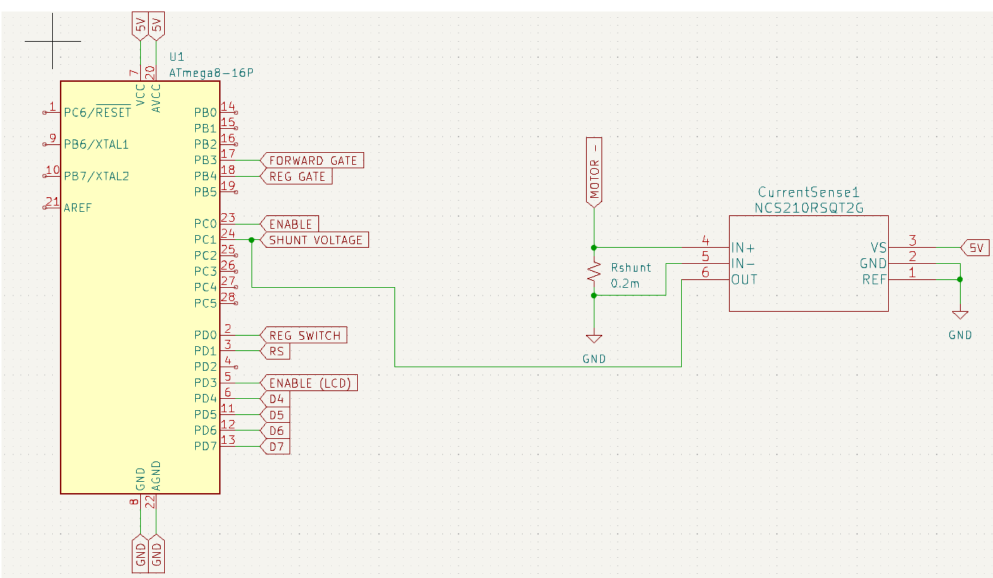


Figure 2.4 Microcontroller Inputs, Outputs, and External Circuitry

The “Enable” signal in Figure 2.4 represents the digital enable signal sent to the boost converter. This signal is straightforward. When the regeneration switch is high, Enable is high. The reverse is also true.

The “Forward Gate” and “Reg Gate” signals represent the switching signals sent to the project’s protection circuit. When the regeneration switch is low, Forward Gate is high, and Reg Gate is low. When the regeneration switch is high, the reverse is true.

Though all three of these signals have simplistic behavior, there is one more implemented behavior to highlight. When the microcontroller detects that the regeneration switch has switched from low to high or high to low, it implements a 10ms delay on the signals meant to go high. Thus, there is a 10ms delay from when high signals are driven low to when low signals are driven high. This behavior is meant to make it such that the forward mode and regenerative mode are never active at the same time. It gives the protection and boost circuits time to catch up to the expected behavior of the bike, ensuring the safety of the user. The delay time of 10ms was also chosen to eliminate the bouncing effects of the regeneration switch. The code begins executing when the switch input is detected to have changed. Due to the relatively longer timescale of this delay, the switch has been completely debounced by the time the block of code has finished executing. This delay behavior will become clearer in Section 3.3.

The more complex secondary function of the microcontroller is to monitor the voltage seen across the shunt resistor, process that data, and display the processed data on an LCD screen. The ground of the motor is connected to true ground using a shunt resistor of resistance 0.2 mΩ. The voltage seen across this resistor is fed into the input of a current sense amplifier with a gain of 200. The output of this amplifier is then seen as the input signal “Shunt Voltage” at the microcontroller.

This voltage is then processed by the microcontroller to arrive at a running calculation of ampere-hours regenerated over the course of one ride. Equation 2.2 is modified slightly to accomplish this task, though it still has the same end goal. The microcontroller first takes one voltage sample and records the time at which it was taken. It then takes another voltage sample, also recording the time at which it was taken. Both samples are converted to current by dividing the voltage by the shunt resistance (0.2 mΩ) and the gain (200 V/V). A conversion is also done to express the analog voltage reading in its appropriate numerical value. This block of code, as well as the rest of the microcontroller code, is shown in Appendix B.

The microcontroller utilizes the trapezoid method of integration. Equation 2.2 illustrates this method. The current values are added together, multiplied by the time difference between samples, and then multiplied by ½. Appropriate conversions are also made to express times taken in milliseconds as hours. The result of this calculation is then added to a running total of regenerated current over time.

$$\frac{1}{2} * (current1 + current2) * (time2 - time1) \quad (2.2)$$

This running total is what is displayed on the LCD. In Figure 2.5, the outgoing signals to the LCD are "RS" and "Enable (LCD)", which are control signals, and "D4", "D5", "D6", and "D7", which are data signals. Interfacing with the LCD required only the inclusion of the Liquid Crystal library and will not be discussed in detail as it is relatively low level. Appendix B contains the full contents of the microcontroller code.

Finally, the processed data is sent to the LCD. The input pins can be seen in Figure 2.5. Register Select (RS) and Enable signals are sent from the microcontroller. Read/Write (R/W) is tied to ground, since data only ever needs to be written to the LCD. Only four of the eight data pins were necessary to transmit data. These are D4-D7. VDD and VSS are power and ground, respectively. LED (+) and LED (-) are power and ground to the LEDs in the LCD, respectively. Finally, an acceptable contrast (V0) was obtained when the contrast pin was tied to ground. A variable contrast can be obtained using a potentiometer, but this contrast proved to be readable.

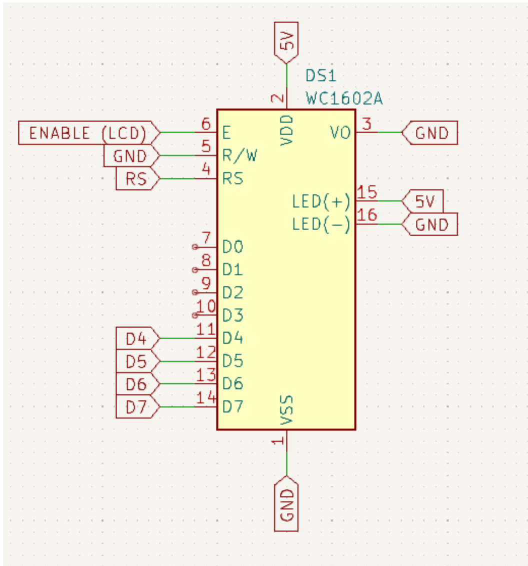


Figure 2.5 LCD Inputs

3. Design Verification

Chapter 3 will discuss the testing of each of the blocks. A full Requirement and Verification table is listed in Appendix C.

3.1 Power Subsystem

The verifications for the power subsystem were relatively simple. The most important part of the power subsystem was providing steady power to the LCD display, the microcontroller, the current sense chip, and the regen button. This power was supplied by converting our 24 V battery into a 5 V output using a linear regulator. The output of the linear regulator was measured while it was powering all these devices to ensure steady operation. It was also tested under a variety of input conditions to ensure proper operation, and unsurprisingly it performed perfectly, as shown below in Table 3.1. The most important requirement for this linear regulator was simply that it wouldn't dip below 3.3 V, as that was the turn on voltage for the systems powered by it, and fortunately it never dropped below 4 V.

Table 3.1 5 V Linear Regulator Output

Input (V)	Average (V)	Peak-to-Peak (V)
5	5.076	1.7
10	5.108	1.7
15	5.097	1.7
20	5.097	1.7
25	5.087	1.7

The other important part of the power subsystem was the boost converter. The most important part of our project, regenerative braking, required a boost converter that could safely and sufficiently direct power to the battery from the motor. Our testing of this converter admittedly should have utilized more load conditions, as it was tested under no load; however, the boost ultimately was able to recover power from the motor, as verified by our microcontroller. Regardless, under no load conditions, our first boost board performed excellently, with an average output ranging from 29.74-29.8 V. However, this board was compromised when a probe shorted the supply voltage to ground. This resulted in a last-minute replacement being recreated, but the replacement was made so quickly that a 16 V and 50 V capacitor were switched, resulting in poor output behavior.

This resulted in a boost converter that only reached the proper output voltage around 16V, which was undesirable. This behavior is shown in Figure 3.1 below.

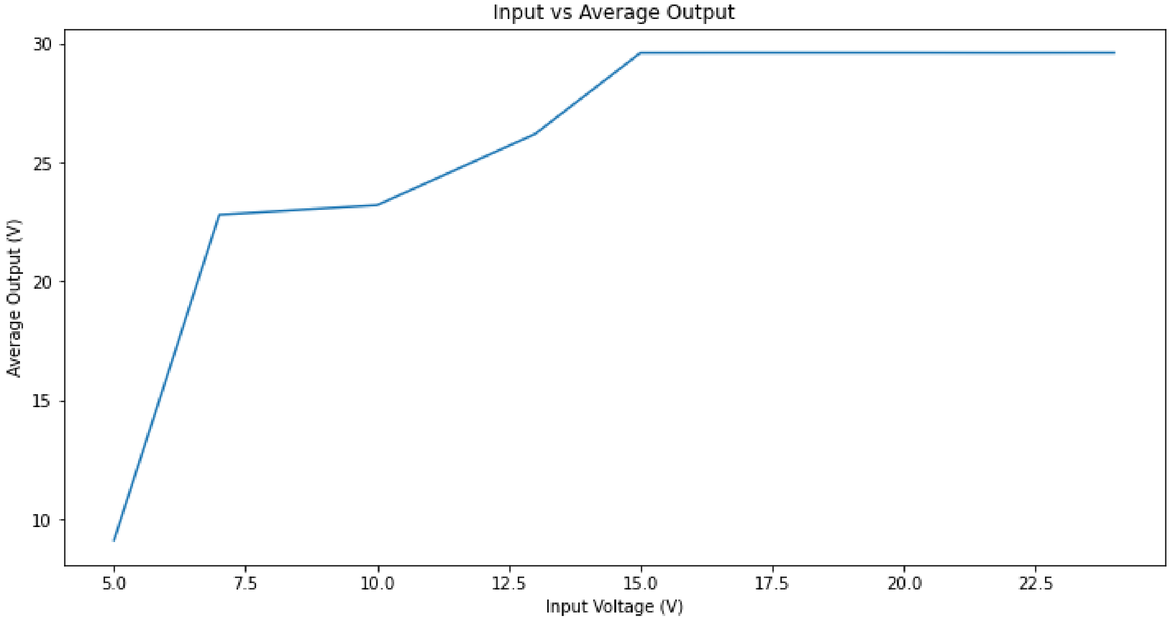


Figure 3.1 Output of Final Boost Circuit

Ultimately, the power subsystem succeeded in providing power to its necessary components, and it accomplished its job successfully.

3.2 Motor Subsystem

The primary role of the motor system was to drive the motor in forward mode and measure the current coming out of the motor in regenerative mode. The motor reliably increased speed in proportion to throttle input, and the motor never activated without throttle input. Furthermore, the bike was able to roughly reach a speed of 10 mph when tested in the ECEB hallway, but not much numerical data was collected on this matter.

We did ensure that the current reading from the motor was accurate, however, as this was a very important part of the regenerative braking process. The input to the microcontroller was tested at voltages similar to the output of the current sense chip. The input was tested at a constant voltage of 0.05 and 0.1V as well as a square wave and sine wave of 0.2 V amplitude. Throughout these tests, the microcontroller calculated value had an error percentage in the range of 0.12-4.33%, which proved that we could accurately measure power recovered from the motor.

3.3 Control Subsystem

The verification for the Enable signal generated by the microcontroller was straightforward. When the regeneration switch was on, Enable read high according to both multimeter and oscilloscope probes. When the regeneration switch was off, Enable read low. During boost output tests, the output voltage increased when the regeneration switch was pressed. Output voltage also decreased to the input voltage when the regeneration switch was flipped to the off position. Since this was a purely digital signal that mirrored an input (with slight modifications for delay behavior), further testing was not required.

The switching signals were tested using oscilloscope probes. In Figure 3.2, the regeneration switch was initially set low. Thus, the Forward Gate signal read high, and the Reg Gate signal read low. When the switch was flipped high, the Forward Gate signal immediately switched low. There was then a period of 10ms where both the Forward Gate and Reg Gate signals were low. After the 10ms period ended, the Reg Gate signal was forced high. This behavior aligns with the desired delay behavior discussed in Chapter 2.

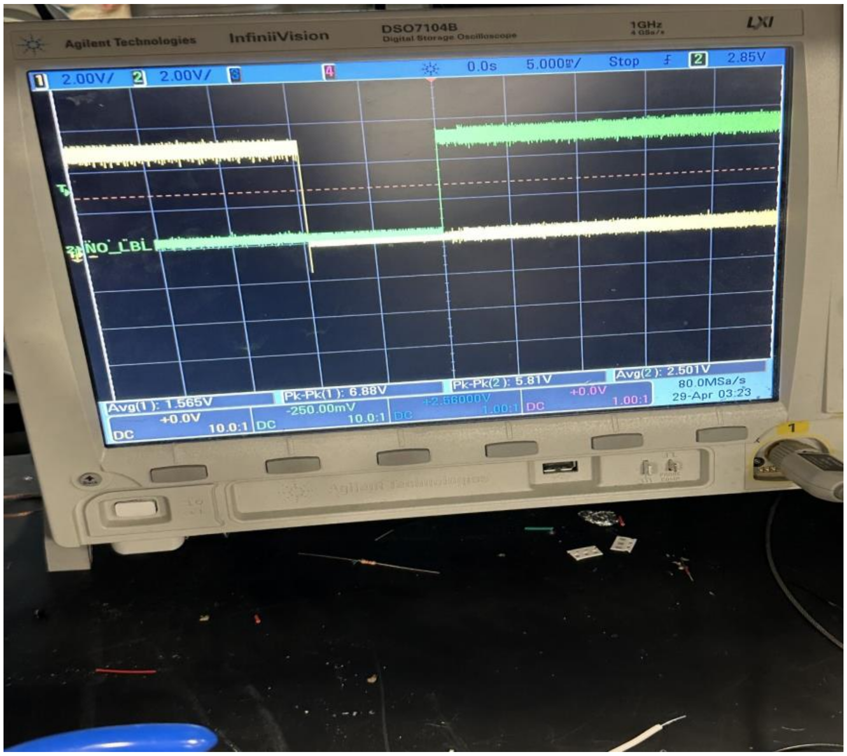


Figure 3.2 Gate Signal Verification

Of course, the converse was also tested, with the regeneration switch being initially high and then being flipped low. This behavior was again observed. Finally, this behavior also holds true for the Enable signal. It is less necessary to delay the Enable signal from switching from low-to-high, but it can add a slight additional element of safety to the project. Delaying the Enable signal also does not change the overall behavior of the bike, since the regeneration circuit will not be activated until the Reg Gate is forced high.

With these tests, the microcontroller fulfills its first two requirements, listed in Appendix C. It responds to and finishes its switching behavior within the one second window required of it.

The Enable and switching signals generated by the microcontroller were digital signals, and thus simple to test. However, the secondary data collection and processing done by the microcontroller is an analog-to-digital process. Since it was important to ensure accuracy, more rigorous tests had to be performed.

Measurement accuracy and LCD testing were performed by feeding either a constant or variable voltage into the Shunt Voltage pin. The function of the voltage was known and recorded, and tests were allowed to run for either ten seconds or one minute. Regenerated current over time in ampere-hours (Ah) was calculated using Equation 2.1 using the voltage function and time elapsed. The Ah reading on the LCD was recorded after the set time had elapsed. Initial testing was performed in TinkerCad using a simulated Arduino Uno board (similar chipsets allow this) to increase the speed of testing and to fix any undesired behaviors caused by the code before physical testing began. All test results displayed in this report were recorded using the physical microcontroller. A list of these tests is shown in Appendix D to save space. Each of the tables contains the observed Ah reading, the calculated Ah reading, and the error between those readings.

Since all testing resulted in outputs within our 5% tolerance and these outputs were displayed on the LCD, these requirements laid out in Appendix C were also met.

Finally, we want to note that there is a requirement that the control subsystem must stay below 125 °C. While the microcontroller is rated for a maximum of this temperature, there is no chance that the electric bike will ever get this hot. The only possible way for this to be accomplished is if the battery fails catastrophically, which our bike has protection against. As a result, this requirement was removed from our Requirements and Verifications table. Thus, the control subsystem exhibits all desired behavior, and meets all conditions described in Appendix C.

4. Costs

4.1 Parts

This section will detail the individual and bulk costs of the components used in this project. Table 4.1 provides this information.

Table 4.1 Parts Costs

Component	Part Number	Manufacturer	Quantity (per unit)	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
Battery	B0B7DYN236	WOGQX	1	103.00	103.00	103.00
Motor	MY1020	Hi-Gear	1	72.00	72.00	72.00
Motor Controller and Throttle	4xcw9dgto1-01	Vbestlife	1	27.58	27.58	27.58
Microcontroller	ATMEGA8A-PU	Microchip Technology	1	3.36	2.73	3.36
Shunt Resistor	PU5931JKH130U2L	YAGEO	1	1.43	0.65	1.43
Linear Regulator	S-1142B50I-E6T1U	ABLIC	1	1.60	0.73	1.60
LCD Display	C162A-BW-LW65	Focus LCDs	1	6.27	4.85	6.27
Boost Converter	LM51561H	Texas Instruments	1	2.01	0.97	2.01
Regenerative Braking Switch	VGEBY2wngre41ph	VGEBY	1	6.74	6.74	6.74
Current Sense	NCS210RSQT2G	Onsemi	1	0.93	0.40	0.93
Capacitors, Inductors, Resistors, etc (Miscellaneous)	-	-	-	-	-	35.00
XT60 Connectors (Male and Female)	XT60FM55	Elechawk	1	8.99	8.99	8.99
XT60 Y Connectors	43234-1912	Fly RC	1	8.99	8.99	8.99
Bike (provided by user)	-	-	1	0.00	0.00	0.00
Total	**-**	**-**	**-**	242.90	237.63	242.90

4.2 Labor
We will estimate our hourly wage to be \$30/hr, and we will work 15 hours a week each for the main 12 weeks of this course.

$$\$30/hr \times 2.5 \times 15\,hrs \times 12\,weeks \times 3\,workers = \$40500$$

For the machine shop, we will estimate a higher wage of 45\$/hour, and we will assume it took 1 worker 4 hours to design and mount our friction wheel.

$$\$45/hr \times 2.5 \times 4hrs \times 1\,worker = \$450$$

This results in a total project cost of \$41,192.90.

5. Conclusion

5.1 Accomplishments

Our e-bike kit, neglecting cost of labor, ultimately cost much less than traditional e-bikes with regenerative braking capabilities. We succeeded in creating a kit that allowed forward motor activation on a regular bike, and the bike could recover power using a regenerative braking button.

5.2 Uncertainties

Our e-bike ultimately failed to successfully integrate the forward mode and regenerative mode properly. We incorrectly used power MOSFETs in our protection circuit, which prevented complete isolation of the boost circuit and the ESC. Furthermore, we implemented our design using a brushed DC motor and a friction wheel. This design is not only inefficient but adds additional wear on the tire of the bicycle.

5.3 Ethical considerations

Due to the potential risks associated with the battery being used for this project, we as a team must take care to minimize those risks, as stated in Section I.1 of the IEEE Code of Ethics [4]. We will take care to ensure that the batteries do not receive more charge than they can hold. During assembly, we must also take care not to drop or damage the battery in any way, as this can be unsafe to users. Charging at cold temperatures and ensuring that the battery does not operate above its rated temperature will also be important considerations. Charging below 0C will also be a consideration for the end user, as this may render our finished project unsafe [5]. Our regenerative braking system provides ample braking power, but manual brakes should remain on any bike this kit may be installed on. When using a friction wheel design, rear tire treads should be checked often, as worn-down tires can result in a loss of traction. To continue, this kit is not designed to function in wet environments, and doing so could damage both the electronics of the bike and the user. This information should be provided to anyone intending to use this kit.

5.4 Future work

Given sufficient time, this system could be better optimized for power efficiency and mechanical stability. A hub motor should be utilized, as it would reduce the weight of the overall kit as well as the wear and tear on the bike tire. A system of relays should be utilized with control signals in order to properly isolate the boost circuit and the ESC. Proper housing for the PCB and better mounting mechanisms could be used for the battery and motor controller as well, which would ultimately reduce the chance of damage to the system.

References

[1] "Over 450 electric bikes compared! What does an ebike cost?," *eBikesHQ.com*, 25-Oct-2022. [Online]. Available: https://ebikeshq.com/cost-of-an-ebike/. [Accessed: 08-Feb-2023].

[2] M. Toll, "The truth: How far can an electric bicycle really go on a single charge?," *Electrek*, 19-Mar-2022. [Online]. Available: https://electrek.co/2020/06/12/how-far-can-an-electric-bicycle-really-go-on-a-charge/. [Accessed: 08-Feb-2023].

[3] "Do bike brake pads get old?," *Cycling Vitality*, 07-Aug-2022. [Online]. Available: https://cyclingvitality.com/do-bike-brake-pads-get-old. [Accessed: 08-Feb-2023].

[4] "IEEE code of Ethics," *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 08-Feb-2023].

[5] "Preventing fire and/or explosion injury from small and wearable lithium battery powered devices," 20-Jun-2019. [Online]. Available: https://www.osha.gov/sites/default/files/publications/shib011819.pdf. [Accessed: 09-Feb-2023].

[6] "Illinois State Electric Bike Laws," *EVELO*, 30-Aug-2022. [Online]. Available: https://evelo.com/blogs/ebike-laws/illinois#:~:text=Class%201%3A%20eBikes%20equipped%20with,the%20speed%20of%2020%20mph. [Accessed: 21-Feb-2023].

[7] "TMS320F28035PAGQ active," *TMS320F28035-Q1 | Buy TI Parts | TI.com*. [Online]. Available: TMS320F28035-Q1 | Buy TI Parts | TI.com [Accessed:21-Feb-2023].

[8] "Universidad politécnica salesiana sede cuenca - ups," Jun-2021. [Online]. Available: https://dspace.ups.edu.ec/bitstream/123456789/20506/1/UPS-CT009183.pdf. [Accessed: 22-Feb-2023].

[9] D. Torres and P. Heath, "Regenerative braking of BLDC Motors - Microchip Technology." [Online]. Available: https://ww1.microchip.com/downloads/en/devicedoc/regenerative%20braking%20of%20bldc%20motors.pdf. [Accessed: 22-Feb-2023].

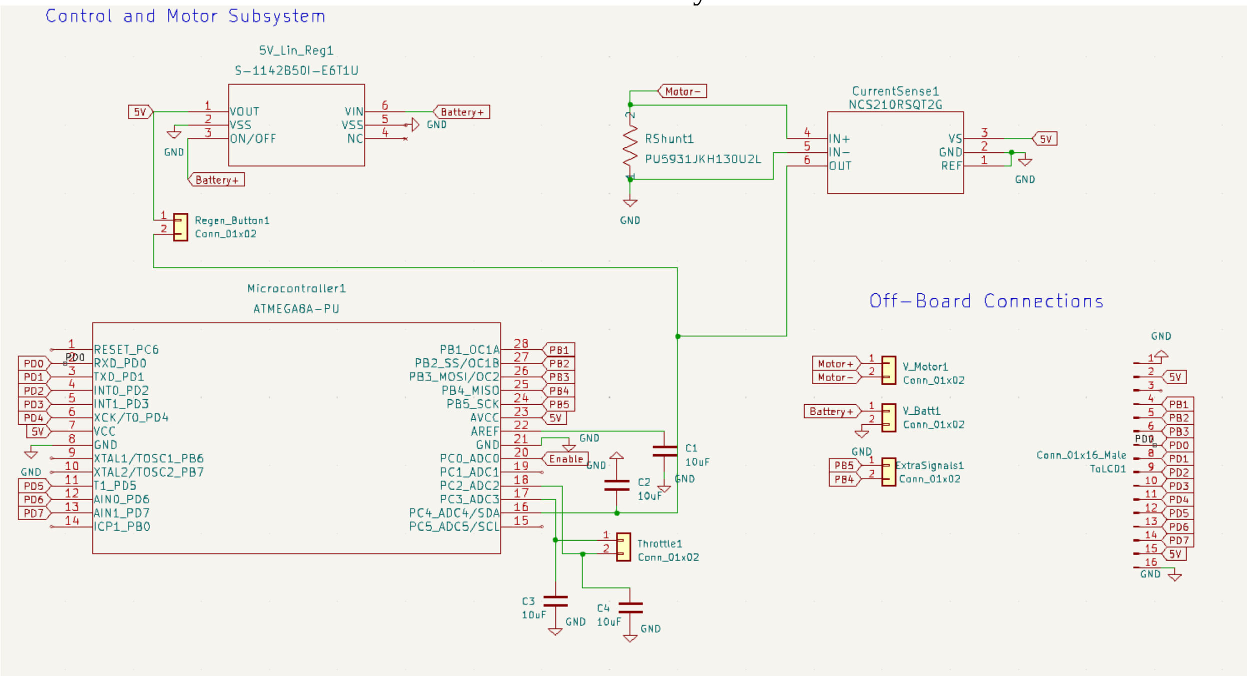
[10] A. Sharma, R. Mishra, A. K. Yadav, and A. Phukan, "Bidirectional DC-DC converter for incorporating regenerative braking in e-bikes," *2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2018.

[11] "ISL81801 Datasheet," *Renesas Electronics Corporation*. [Online]. Available: https://www.renesas.com/us/en/document/dst/isl81801-datasheet. [Accessed: 24-Feb-2023].

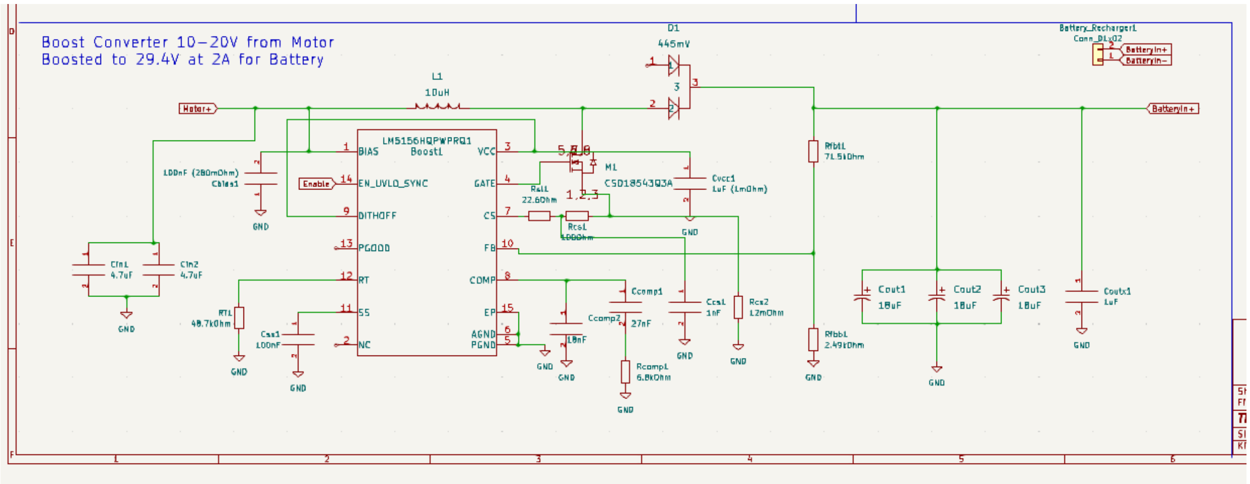
[12] "Atmega8a Data sheet summary - microchip technology." [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Microchip%208bit%20mcu%20AVR%20ATmega8A%20data%20sheet%20summary%2040001991A.pdf [Accessed: 24-Mar-2023].

Appendix A Complete Circuit Diagrams

Control and Motor Subsystem Circuit



Boost Converter Circuit



Appendix B Microcontroller Code

```
#include <LiquidCrystal.h>

//enable pin for regen
#define EN_PIN PIN_PC0

//regen button pin
#define REG_PIN PIN_PD0 //maybe debounce?

//MOSFET switching
#define M1_PIN PIN_PB3
#define M2_PIN PIN_PB4

//analog voltage reading from shunt
#define VS_PIN A1

//RS, EN, D4, D5, D6, D7k
#define RS_PIN PIN_PD1
#define RW_PIN PIN_PD2
#define E_PIN PIN_PD3
#define D4_PIN PIN_PD4
#define D5_PIN PIN_PD5
#define D6_PIN PIN_PD6
#define D7_PIN PIN_PD7
LiquidCrystal lcd(RS_PIN, E_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN);

//EXTRA power and ground pins
#define CONTRAST PIN_PC4

//variables that will hold voltage samples
double data1 = 0.0;
double data2 = 0.0;

//time at which these samples were taken
unsigned long data1time;
unsigned long data2time;

//final amp hour reading
double regenpwr = 0;
```

```
void setup() {
  // put your setup code here, to run once:

  pinMode(REG_PIN, INPUT);
  pinMode(EN_PIN, OUTPUT);
  pinMode(M1_PIN, OUTPUT);
  pinMode(M2_PIN, OUTPUT);

  digitalWrite(EN_PIN, LOW); //enable pin pulled low during startup
  digitalWrite(RW_PIN, LOW); //grounded throughout

  digitalWrite(CONTRAST, LOW); //grounded throughout

  //normal switching mode on startup
  digitalWrite(M2_PIN, LOW);
  delay(10);
  digitalWrite(M1_PIN, HIGH);

  lcd.begin(16,2);
  lcd.setCursor(6,0);
  lcd.print("Ah");
  lcd.setCursor(0,0);

}

void loop() {
  // put your main code here, to run repeatedly:

  //setting a delay time, protection against both circuit operating continuously
  int delay_time = 10;

  //if regen switch is on and enable is low, drive enable high
  if(digitalRead(REG_PIN) == HIGH && digitalRead(EN_PIN) == LOW) {
    digitalWrite(M1_PIN, LOW);
    delay(delay_time);
    digitalWrite(EN_PIN, HIGH);
    digitalWrite(M2_PIN, HIGH);
  }

  //if regen switch is off and enable is high, drive enable low
  else if(digitalRead(REG_PIN) == LOW && digitalRead(EN_PIN) == HIGH) {
    digitalWrite(EN_PIN, LOW);
    digitalWrite(M2_PIN, LOW);
    delay(delay_time);
    digitalWrite(M1_PIN, HIGH);
  }

  //Important note: the analog in pins read voltages at numbers between 0 and 1023
  //These correspond to voltages between 0 and 5
  //changing while to if, loop was producing unwanted behavior
  if((digitalRead(REG_PIN) == HIGH) && (int32_t(analogRead(VS_PIN)) >= 0)){
      //transfer last sample value and time to new variables
      data1 = data2;
      data1time = data2time;

      //sample is current, voltage at the shunt divided by the resistance of the shunt, 200 represents gain
      data2 = (double(analogRead(VS_PIN))*5.0 /1024.0) / (0.0002*200);
      data2time = millis();

      regenpwr += 0.5 * (data2 + data1) * (data2time - data1time)*1/3600000;
  }
  //reset data and time it is taken at
  else{
    data2 = 0;
    data2time = millis();
  }
  lcd.print(regenpwr);
  lcd.setCursor(0,0);
}
```

Appendix C Requirements and Verification

Power Subsystem Requirements

Requirements	**Verification**
• Convert regenerative power from the motor into 29.4V±1.0V to recharge the motor.	• Spin the motor wheel at various speeds from 5-20 mph and activate regenerative braking. • Place an oscilloscope probe on the output to the battery • Measure the output voltage of the buck-boost convertor with an oscilloscope and verify correctness.
• Subsystem must provide 5V ± 1V to the microcontroller, regen button, and LCD display.	• Place a voltmeter probe on linear regulator output. • Read the voltage with all loads connected and active. • Verify that the supply voltage is present and within parameters.

Motor Subsystem Requirements

Requirements	Verification
• Measure current output from the motor when regenerating over time. Accuracy must be within ± 5% of a multimeter reading	• The current must be reliably measured over time, and that data will be stored in the microcontroller's permanent storage. This data will be compared to the readings from a multimeter in the lab. • Set up a shunt resistor and run 1A current through it. • Display the current reading, either through the LCD or through an output on the microcontroller. • Compare reading to that taken by an oscilloscope probe.
• Motor voltage changes reliably with the throttle input. Motor voltage is within ± 5% of the expected motor voltage based on the duty cycle.	• Test the circuit while not on the bike. • Supply switching signals to the switching converter, note what output should be produced in buck configuration. • Expected and experimental voltage values should be compared. • Calculate the duty cycle associated with expected voltage outputs of 5V, 10V, and 15V. • Measure the actual voltage that the duty cycle produces. • Verify output with a multimeter. • Expected voltage must be within 5%.
• Motor does not activate without throttle activation. If twist throttle input is within 0.5V of resting position voltage, the motor should not be on.	• Measure the output voltage from the throttle when it is at 0.5V from rest using an oscilloscope and ensure that the motor isn't activated. • Measurement will be taken using a multimeter. • Ensuring the motor is not activated translates to motor rpm being 0, or very near 0.

Control Subsystem Requirements

Requirements	**Verification**
• Microcontroller must recognize the regenerative braking "ON" signal and change from normal switching mode to regenerative braking switching mode within 1s.	• Off-bike testing: Supply regenerative braking signal to microcontroller. • Read switching signals on oscilloscope. • If they change to the correct switching signals within 1s, the microcontroller passes this step of verification. • On-bike testing: Press and hold the regenerative braking button once the bike has reached a speed of 10mph. • Project passes the test if the user can feel the slowing down of the bike within 1s.
• Microcontroller must recognize the regenerative braking "OFF" signal and change from regenerative braking switching mode to normal switching mode within 1s.	• Off-bike testing: Supply regenerative braking signal to microcontroller, then switch off. • Read switching signals on an oscilloscope. • If they change to the correct switching signals within 1s, the microcontroller passes this step of verification. • After pressing and holding the regenerative braking button, apply a steady throttle input. • Passes if the bike begins to accelerate within 1s.

Appendix D Microcontroller LCD Accuracy Tests

This appendix lists a variety of voltage tests performed to ensure microcontroller accuracy. For each test, the applied voltage will be given. Observed and expected Ah readings are recorded, and the error between the measurements is calculated.

Function: Constant **V: 100 mV**	10 seconds	60 seconds
Recorded	1.44	8.59
Expected	1.39	8.33
Error	1.45	1.68

Function: Square Wave **Vpp: 100 mV** **Offset: 50mV** **Frequency: 0.2 Hz**	10 seconds	60 seconds
Recorded	0.68	4.10
Expected	0.69	4.17
Error	1.45	1.68

Function: Sine Wave **Vpp: 200 mV** **Offset: 100mV** **Frequency: 0.2 Hz**	10 seconds	60 seconds
Recorded	1.33	7.93
Expected	1.39	8.33
Error	4.32	4.80